KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 138–144

COLLABORATIVE SELECTION FOR EVOLUTIONARY ALGORITHMS

ANCA GOG [(1)] AND D. DUMITRESCU [(2)]

Abstract. A new selection scheme for evolutionary algorithms is proposed. The introduced selection operator is based on the collaboration between individuals that exchange information in order to accelerate the search process. The NP-hard Travelling Salesman Problem is considered for testing the proposed approach. Numerical experiments prove the efficiency of the proposed technique, compared with the most popular selection operators used within evolutionary algorithms.

1. Introduction

A new selection operator is proposed in order to improve the search process of the evolutionary algorithms. Unlike the standard evolutionary algorithm, in the proposed approach each individual has information about its best related individual. The new collaborative selection operator is based on this extra information that each individual has and tends to favor the fittest individuals within each group of individuals having a common best ancestor.

The proposed Collaborative Selection (CS) operator is compared to the most popular selection operators. Several instances of the NP-hard Travelling Salesman Problem (TSP) are considered in order to prove the efficiency of the proposed operator.

The paper is organized as follows: Section 2 presents some of the existing selection operators; Section 3 describes the new proposed Collaborative Selection operator; Section 4 contains experimental results and there are conclusions in Section 5.

2. Selection Operators

The purpose of the selection procedure is to choose from the current population the set of individuals (parents) that will be used to create the next generation. The

2000 *Mathematics Subject Classification.* 68T20, 68T99.
Key words and phrases. Collaborative Selection, Travelling Salesman Problem, Combinatorial Optimization.

©2007 Babeş-Bolyai University, Cluj-Napoca

choice of which individuals are allowed for reproducing determines which regions of the search space will be visited next. This choice is often the result of a trade-off between exploration and exploitation of the search space. Some of the most popular selection operators [1] are briefly described in what follows.

Roulette Selection is used to generate a uniform probability distribution. This mechanism ensures the selection of the individual x^i with probability:

$$p_i = \frac{f(x^i)}{F}, i = 1, 2, \ldots, n,$$

where F is the total fitness of the population, defined as the sum of all individuals' fitness and $f(x^i)$ is the fitness of the individual x^i.

In *Linear Ranking Selection* at each generation the individuals are sorted according to their fitness and a rank is assigned to each individual in the sorted population. The selection probabilities of the individuals are given by their rank in the population.

In *Tournament Selection* N individuals are chosen from the population in order to produce a tournament subset of chromosomes. The best chromosome in this subset is then selected.

Best Selection operator selects the best chromosome determined by fitness. If there are two or more chromosomes with the same best fitness, one of them is chosen randomly.

3. Collaborative Selection Operator

In evolutionary algorithms, one individual (or chromosome) encodes a potential solution of the problem and is composed by a set of elements called genes. Each gene can take multiple values called alleles. In the proposed collaborative approach an individual has extra information regarding its best related individual, the so-called *LineOpt* (related individuals refer to all individuals that have existed in one of the previous generations and have contributed to the creation of the current individual: its parents, the parents of its parents, and so on).

Selection operator should provide a good equilibrium between the exploration and the exploitation of the search space. On one hand, selection has to provide high reproductive chances to the fittest individuals; on the other hand, selection must preserve population diversity in order to explore all promising search space regions.

Selection operator chooses which individuals should enter the mating pool in order to be subject of recombination. On this purpose, the proposed selection operator called *Collaborative Selection (CS)* uses the information about the *LineOpt* of each individual. This selection is essentially rank-based.

Let us assume that the current population *P(t)* has n individuals:

$$P(t) = \{x^1, x^1, \ldots, x^n\}.$$

Let f be the fitness function. Within Monte Carlo selection mechanism [1] the selection probability of the individual x_i is the number p_i defined as:

$$p_i = \frac{f(x^i)}{F}, i = 1, 2, \ldots, n,$$

where F is the *total fitness* of the population.

In order to prevent an individual or a group of high-fit individuals from dominating the next generation the introduced CS operator uses rank information.

All individuals within the current population are grouped by their *LineOpt*. Considering the current population *P(t)*, clusters $A_1, \ldots, A_k, k \leq n$, are formed according to the rules:

(i) the clusters $A_1, \ldots, A_k, k \leq n$ represent a partition of *P(t)*:

$$(a) \quad A_i \neq \phi, 1 \leq i \leq k,$$

$$(b) \quad \bigcup_{i=1}^{k} A_i = P(t),$$

$$(c) \quad A_i \bigcap A_j = \phi, 1 \leq i \leq k, 1 \leq j \leq k, i \neq j.$$

(ii) all the individuals that belong to the cluster $A_i (1 \leq i \leq k)$, have the same *LineOpt*:

$$\mathit{LineOpt}(x^i) = \mathit{LineOpt}(x^j),$$
$$\forall x^i, x^j \in A_l, 1 \leq l \leq k.$$

(iii) every two different clusters $A_i, A_j (1 \leq i \leq k, 1 \leq j \leq k, i \neq j)$ have a different *LineOpt*:

$$\mathit{LineOpt}(x^i) \neq \mathit{LineOpt}(x^j),$$
$$\forall x^i \in A_i, x^j \in A_j, 1 \leq i \leq k, 1 \leq j \leq k, i \neq j.$$

Let us suppose that there are two individuals in two different clusters having the same fitness value:

$$x^i \in A_i, x^j \in A_j, 1 \leq i \leq k, 1 \leq j \leq k, i \neq j,$$
$$f(x^i) = f(x^j).$$

Furthermore, let us suppose that x^i is the fittest individual in the cluster A_i and that x^j is the worst individual in the cluster A_j. The Monte Carlo technique assigns the same probability of being selected to individuals having the same fitness. The CS operator favors the individual x^i by assigning it a higher probability of being selected than the probability of the individual x^j, even if both of them have the same fitness. The reason is the fact that x^i is the fittest individual of the cluster A_i, while x^j is the worst individual of the cluster A_j. The goal of the proposed strategy is to favor the selection of the fittest individuals within each cluster, i.e. within each group of individuals having the same *LineOpt*.

This goal is achieved by modifying the selection probability p_i. Within CS individuals in each cluster are ranked according to their relative fitness. A pure rank-based selection scheme (like tournament) may be used. A different approach modifies the selection probability according to the rank. In this case the selection probability p_i of an individual is modified according to the rank of the individual in its cluster but in such a way that the sum of renormalized probabilities p_i remains 1.

Renormalized probability is computed for each individual according to its membership to a cluster and its rank in that cluster. Let us suppose that the cluster $A_i (1 \leq i \leq k)$, contains the individuals $x^j, (1 \leq j \leq |A_i|)$. For each individual x^j the relative rank in the cluster A_i is the number pc_j computed as:

$$pc_j = \frac{rank(x^j)}{\sum\limits_{l=1}^{|A_i|} l}, j = 1, 2, \ldots, |A_i|.$$

The selection probability of the individuals x^j belonging to the cluster A_i is modified according to their relative ranks. The new probability new_p_j of the individual x^j is computed as follows:

$$new\_p_j = pc_j * S,$$

where S is the sum of the probabilities p_j for all individuals x^j from the cluster A_i. This way, the sum of the new probabilities for all individuals from a cluster satisfies:

$$\sum_{x^j \in A_i} pc_j = 1.$$

Indeed we may successively write,

$$\sum_{x^j \in A_i} new\_p_j = \sum_{x^j \in A_i} pc_j S = S \sum_{x^j \in A_i} pc_j = S = \sum_{x^j \in A_i} p_j.$$

The sum *1* for the new computed probabilities for all the individuals within the population is conserved.

The new probabilities favor the fittest individuals as well as the fittest individuals within each cluster ensuring that good genetic material already obtained is preserved. This promotes the exploitation of all promising regions discovered during the search process therefore preventing a group of high-fit individuals from dominating the next generation.

4. Experimental results

Evolutionary Computation provides good approximate methods for solving Combinatorial Optimization Problems, especially NP-complete and NP-hard problems [2], [5]. One representative combinatorial optimization problem, namely Travelling Salesman Problem (TSP) [3] is investigated to prove the efficiency of the proposed selection operator. A set of k points in a plane is given, corresponding to the location of k cities. The Travelling Salesman Problem requires finding the shortest closed path that visits each city exactly once. The problem can be formalized as follows:

A set of k cities

$$C = \{c_1, c_2, \ldots, c_k\}$$

is given. For each pair

$$(c_i, c_j), i \neq j,$$

let

$$d(c_i, c_j)$$

be the distance between the city c_i and the city c_j. One has to find a permutation π' of the cities

$$(c_{\pi'(1)}, \ldots, c_{\pi'(k)}),$$

such that

$$\sum_{i=1}^{k} d(c_{\pi'(i)}, c_{\pi'(i+1)}) \leq \sum_{i=1}^{k} d(c_{\pi(i)}, c_{\pi(i+1)}),$$

$$\forall \pi \neq \pi', (k+1 \equiv 1).$$

This problem can be defined as the search for a minimal Hamiltonian cycle in a complete graph.

The simplest evolutionary approach of this NP-hard problem is outlined in what follows. A potential solution for the problem (a chromosome) is a string of length k that contains a permutation π of the set

$$\{1, \ldots, k\},$$

and represents the order of visiting the k cities. A chromosome is evaluated by means of a fitness function f that needs to be minimized:

$$f : S \to \Re^+, f(\pi) = \sum_{i=1}^{k} d(c_{\pi(i)}, c_{\pi(i+1)}),$$

$$(k+1 \equiv 1),$$

where S represents the search space of the problem, i.e. the set of all permutations π of the set

$$\{1, \ldots, k\}.$$

Thus, the fitness of a chromosome is the length of the closed path that visits the cities in the order specified by the permutation π.

A Standard Genetic Algorithm (SGA) with OX recombination and inverse mutation [1] is considered for numerical experiments. Several TSP instances taken from TSPLIB are investigated [4]. For the considered problems, SGA is applied with Collaborative Selection and with all selection operators described in Section 2: Roulette Selection, Linear Rank Selection, Tournament Selection and Best Selection. Table 1 and Table 2 contain the results obtained after 10 runs of the SGA with all considered selection operators. Results regard the average solution obtained after 100 and after 500 generations. The best values obtained are bolded in both tables.

Table 1. Average results obtained after 10 runs of SGA (after 100 generations) with all considered selection operators.

TSP instance	**Roulette Selection**	**Linear Rank Selection**	**Tournament Selection**	**Best Selection**	**Collaborative Selection**
EIL51	878	745	**753**	**753**	755
ST70	2063	1707	1726	1713	**1672**
PR76	328475	290645	288150	285725	**278838**
EIL76	1439	1281	1294	1276	**1271**
KROA100	95431	82792	**81908**	84672	82225
KROB100	93425	79438	81398	86447	**78462**
KROC100	95164	83840	81577	83297	**80989**
KROD100	92065	81609	**79202**	81549	81604
KROE100	97615	84903	82678	87209	**79437**
EIL101	2053	1890	1862	1838	**1807**

The test results indicate the acceleration of the search process when using CS, especially in the first generations of the algorithm, compared with all the other selection operators. In the latest stages of the algorithm, the only selection operator that outperforms CS is the tournament selection, but CS outperforms the other selection operators in most of the cases.

5. Conclusions and Further Work

A new collaborative selection operator (CS) for evolutionary algorithms has been proposed. CS is using extra information regarding the best ancestor of each individual obtained so far by the search process. Numerical experiments, for which several instances of TSP have been used, have proved that the proposed selection outperforms most of the existing selection operators by accelerating the search process.

TABLE 2. Average results obtained after 10 runs of SGA (after 500 generations) with all considered selection operators.

TSP instance	**Roulette Selection**	**Linear Rank Selection**	**Tournament Selection**	**Best Selection**	**Collaborative Selection**
EIL51	550	514	**498**	499	**498**
ST70	1102	981	**942**	980	962
PR76	187717	158380	**151715**	159378	160502
EIL76	841	763	761	768	**738**
KROA100	51281	41869	**41389**	42315	42765
KROB100	48812	42164	42361	42769	**41996**
KROC100	49882	41344	**40934**	42450	41531
KROD100	48318	40023	**39998**	42179	41579
KROE100	51897	**39871**	41006	42770	40780
EIL101	1205	1039	**1028**	1047	1046

REFERENCES

[1] Dumitrescu, D., Lazzerini, B., Jain, L.C. and Dumitrescu, A., Evolutionary Computation, CRC Press, Boca Raton, FL., 2000.
[2] Blum, C., Roli, A., Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, ACM Computing Surveys, Vol. 35, 268-308, 2003.
[3] Gutin, G., Punnen, A.P., Traveling Salesman Problem and Its Variations, Kluwer Academic Publishers, 2002.
[4] Reinelt, G., TSPLIB - A Traveling Salesman Problem Library, ORSA Journal of Computing, 376-384, 1991.
[5] Alba Torres, E., Khuri, S., Applying Evolutionary Algorithms to Combinatorial Optimization Problems, ICCS 2001, LNCS 2074, Springer Verlag Berlin Heidelberg, 689-698, 2004.

(1) BABES-BOLYAI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
KOGALNICEANU 1,
R0 - 400084 CLUJ-NAPOCA
E-mail address: anca@cs.ubbcluj.ro

(2) BABES-BOLYAI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
KOGALNICEANU 1,
R0 - 400084 CLUJ-NAPOCA
E-mail address: ddumitr@cs.ubbcluj.ro